We present a formulation of non-penetration constraint between pairs of polytopes which accounts for all possible combinations of active contact between geometric features. This is the first formulation that exactly models the body geometries near points of potential contact, preventing interpenetration while not over-constraining body motions. Unlike many popular methods, ours does not wait for penetrations to occur as a way to identify which contact constraints to enforce. Nor do we over-constrain by representing the free space between pairs of bodies as convex, when it is in fact nonconvex. Instead, each contact constraint incorporates all feasible potential contacts in a way that represents the true geometry of the bodies. This ensures penetration-free, physically-correct configurations at the end of each time step while allowing bodies to accurately traverse the free-space surrounding other bodies. The new formulation improves accuracy, dramatically reduces the need for ad hoc corrections of constraint violations, and avoids many of the inevitable instabilities consequent of other contact models. Although the dynamics problem at each time step is larger, the inherent stability of our method means that much larger time steps can be taken without loss of physical fidelity. As will be seen, results obtained with our method demonstrate the effective elimination of interpenetration, and as a result, correction-induced instabilities, in multibody simulations.

Introduction

There is a wide spectrum of applications for simulation, but an understandable bias toward real-time interactivity has bred a culture that leaves concerns of physical accuracy in a dark corner. As a result, applications which require some level of physical fidelity from simulation are left with a relatively underdeveloped set of tools. The major focus of the work presented herein is on improving simulation accuracy, and consequently stability, without significantly increasing computational cost.

This trade-off of speed versus accuracy in simulation is a classic one, and many applications simply do not gain from improved physical accuracy. The gaming community is naturally concerned with methods that are fast and reliable, but not necessarily accurate [Eri04, Ebe10, Mil10], inspiring the term “game physics.” Many games, in fact, intentionally incorporate nonrealistic physics into their play. The graphics community, which is constantly pushing the boundaries of realistic-looking visual effects, does not offer many improvements to physically accurate simulation. Although there are many examples in a long history of “physics based” methods in use in animation [TPB+89, Bar94, FM95, WB97, ESHD05], there is really no need for such methods to do more than approximate some convenient alternative version of reality. Indeed, the goal in many graphics applications is to present the viewer with something that “feels” correct.

Particularly in the field of robotics, physical fidelity is a pressing concern and is especially challenging when simulating systems experiencing intermittent frictional contact, which is the case when robots work in unstructured, cluttered environments. Robotics experiments tend to be exceedingly costly and time consuming, so that the ability to perform re-
alistic experiments in simulation would be a great benefit. Although work has been done regarding validation of simulation [PG96, Ber09, BNAT10, Mac12], the problem of measuring and quantifying the physical accuracy of simulation is an open issue that is challenging to solve. Comparison of a particular simulator’s results with those obtained from a commercial simulator is meaningless without proof of the commercial simulator’s accuracy. Even validation of a simulator with a physical experiment can be misinterpreted when it takes the tuning of several parameters to match a single or small number of experiments. Such an attempt at validation says little, if anything, of the simulator’s ability to predict a different experiment.

Simulation could be a powerful tool if it was shown to be accurate. A simulation requires accuracy in order to make any claims as to its ability to predict behavior in the physical world, yet the growing number of variants on old simulation methods all have at least this in common: they sacrifice physical fidelity at multiple stages in order to achieve computational speed-up. The ideal simulation tool would be both accurate and fast, but when we sacrifice accuracy, we are necessarily sacrificing the usefulness of that tool.

Our new polytopal exact geometric (PEG) contact model presented herein is an important step toward improving the physical fidelity of simulations of polytopal rigid bodies in intermittent contact. In most popular simulation tools, in order to make it easy to formulate the nonpenetration constraints, the shapes of the bodies are effectively altered. Specifically, facets of polytopes are represented as hyper-planes (e.g., faces of polyhedra in 3D are represented as planes). As will be illustrated below, the price for this convenience is very high, as it leads to nonphysical simulation artifacts, such as huge constraint forces that eject bodies from the scene, or worse, cause the simulation’s solvers to fail completely. By contrast, PEG exactly represents the boundaries of polytopal bodies, including their finite extent, and thus avoids these problems. To accomplish this, the PEG contact model is more complicated than the standard contact model. With the same time step sizes, simulation with PEG takes approximately twice as long as simulation with the standard contact model. However, its improved accuracy avoids nonphysical artifacts, including penetration-induced instabilities that can crash the simulation. In our opinion, this speed reduction is a penalty worth paying.

In the remainder of this section, we review multibody dynamics and some of the approaches used in simulation. In section 1, we detail the deficiencies of standard contact modeling practices and begin to derive the PEG contact model by introducing new approaches to cases that standard contact models fail to accurately represent. In section 2, we define the general formulations of three classes of nonpenetration constraints, and provide abstractions of them in order to model contact in a clean way. In section 3, we precisely define contact constraint applicability and feasibility and establish geometric tests to determine which contacts to include in PEG. In section 4, we utilize the abstractions from the previous section in order to define four fundamental contact constraints for spatial problems, which reduce to two when specialized to the planar case. Then we demonstrate how to formulate a time-stepping sub-problem that incorporates PEG. In section 5, we compare the performance of multi-body simulations using a popular time-stepping method with the standard contact model against the same time-stepper with the PEG contact model. Finally, section 6 discusses our results and possibilities for future work.

Previous Work

The two main classes of simulation methods for multi-body systems are known as event-driven and time-stepping. In event-driven methods, the simulation is monitored for impact and separation events and changes between sticking and slipping (see [HWY86]). For systems with small numbers of contacts, this method is preferred, because it uses an adaptive step size to accurately pinpoint and process the contact state transitions. Its downfall is that in systems with many bodies and contact events, the size of the time steps become so small that computation times become impractically large. On the other hand, this is precisely the circumstance in which time-stepping methods have their greatest value. Time-stepping methods use large time steps that can advance the simulation across a number of contact events in one step, such that at the end of the time step Newton’s Laws are satisfied and bodies do not overlap [LN04].

Time-stepping simulation methods utilizing the complementarity problem (defined formally later) were introduced by Moreau [MP88]. In these methods, the data needed to advance the simulation by one time step is obtained as the solution of a complementarity problem, the so-called time-stepping sub-problem. Stewart and Trinkle introduced a modification of this problem that incorporated stabilization of nonpenetration constraints [ST96] and they provided a proof that body trajectories approximated by the solutions to a sequence of these sub-problems converged to a solution of the instantaneous-time formulation as the step size goes to zero [Ste98]. Anitescu and Potra [AP97] included bilateral constraints and showed that the sub-problem with linearized friction cones is always solvable.

The DART [DAR] physics engine uses semi-implicit, velocity-based time-stepping sub-problems, formulated as linear complementarity problems with nonpenetration constraint stabilization. However, because it uses standard contact modeling, it suffers from inaccuracy and instability as discussed above. Chrono::Engine [MHP+13] is built on top of Bullet’s collision detection engine [Cou], which reports penetration depth and gap distance, implying that penetration is not avoided. Across the gamut of the physics engines available today, simulations with bodies modeled as spheres or unions of spheres are popular for benchmarking, especially when there are large numbers of contacts. This is because when simulating polytopes in contact using the standard contact model, simulations fail due to nonphysical, over-constraint imposed on the polygons. These over-constrained situations, henceforth referred to as standard model traps (SM-traps), can cause bodies to obtain huge
nonphysical velocities or complete failure of the simulation.

**Background**

In this section, we will introduce an *instantaneous-time model* of polyhedral bodies in intermittent contact, and then discretize it to obtain the discrete-time counterpart. To simulate the motions of the bodies, the *discrete-time model* (i.e., the time-stepping sub-problem) must be formulated and solved repeatedly, each time advancing the simulation forward in time by the chosen time step. Note that the development herein excludes contact friction in order to maintain a sharp focus on the new contact model, which depends only on body geometry and not friction. Friction can however be easily incorporated by augmenting the time-stepping subproblem with one’s favorite friction model, several of which are detailed in [BET14, Wil14b].

Before formulating PEG and a time-stepping subproblem that employs it, it is important to introduce the complementarity problem [CPS92].

**Complementarity**

The complementarity problem offers a convenient way to model contact because it naturally represents disjunctive conditions. The two key cases in multibody dynamics are: two bodies are in contact OR not in contact and an existing contact is sticking OR sliding. There is a vast body of research on complementarity problems and methods for their solution [BM00, BETC12].

Let $\mathbb{R}^n$ denote $n$-dimensional Euclidean space. Given a mapping $f(z) : \mathbb{R}^n \to \mathbb{R}^n$, the complementarity problem is to find the unknown vector $z \in \mathbb{R}^n$ that satisfies the three conditions


define the complementarity problem

$$
\begin{align*}
  z &\geq 0 \\
f(z) &\geq 0 \\
z^T f(z) &= 0
\end{align*}
$$

where $T$ is the matrix transpose operator. To save space, we may write all three conditions as

$$
0 \leq f(z) \perp z \geq 0,
$$

where $\perp$ represents the orthogonality of condition 3 in equation (1). It is common to refer to all three of these conditions jointly as a *complementarity condition*.

There are many general-purpose solvers available for complementarity problems [FM00] and in particular for linear complementarity problems (LCPs) [CPS92, Erl13]. In an LCP, $f(z)$ is a linear function of $z$, i.e., $f(z) = Az + b$, where $A$ is a known constant square matrix and $b$ is a known constant vector of compatible size. Much theory for LCPs has been developed and used to design reliable solvers. It is known, for example, that if $A$ is positive definite, then the LCP is equivalent to a convex quadratic program and therefore has a unique solution and can be solved efficiently. If $A$ is positive semidefinite, then a solution can be found efficiently also, but some unknowns may not have unique solutions. In multibody dynamics, this case arises only for frictionless systems, and for these, the body motions have unique solutions, but the contact forces do not.

If friction is included in the model, these nice properties are no longer present. Time-steppers using polyhedral friction cones still yield LCPs, but $A$ becomes indefinite and correspondingly, solutions are not unique; they might not exist and when they do, there can be infinitely many. In the special case when all contacts are sticking and Coulomb friction cones are approximated by four-sided pyramids, solutions are known to exist [PT96]. However, the problems are more challenging to solve if the friction model is nonlinear, as is the case for quadratic Coulomb friction, where no existence results are available. Nonetheless, some solvers have success solving the nonlinear complementarity problems (NCPs) that result. We utilize the general-purpose PATH solver for many NCP-based simulations. Work has been done to create specialized solvers for multibody dynamics [Tod11].

**Instantaneous-Time Model**

Consider two bodies $A$ and $B$ as depicted in Figure 1. When the signed gap distance $\psi_i$ is positive (as in the figure), the bodies are separated. When it is zero or negative, they are in contact or interpenetrating. The latter is not allowed, that is, $\psi_i \geq 0$. Generally, a contact between two bodies may be represented as a 5-tuple $C = \{A_{id}, B_{id}, p_d, \hat{n}, \psi_i\}$, where $A_{id}$ and $B_{id}$ are body indices or identifiers for $A$ and $B$ respectively, $p_d$ is the contact point on $A$ in world coordinates, $\hat{n}$ is the contact normal vector in the direction of $A$ onto $B$ by convention. The point of contact $p_d$ on $B$ is available by $p_b = p_d + \psi_i \hat{n}$. The vectors $r_a$ and $r_b$ span from the center of mass of each body to the point of contact on that body. While not required, we use the convention that the origin of a body’s body-fixed reference frame is coincident with its center of mass. Vectors $\mathbf{f}$ and $\mathbf{\bar{f}}$ form an orthonormal basis with $\hat{n}$.

![Fig. 1: Bodies $A$ and $B$ in 3D, and contact vectors in the world frame $\{W\}$.](image_url)
Newton-Euler equation

\[
M(q)\ddot{v} = G(q)\lambda + \lambda_{ext}
\]  

where \( M(q) \) is a block diagonal inertia matrix where the \( j^{th} \) block is the \((6 \times 6)\) inertia matrix of the \( j^{th} \) body

\[
M_j(q) = \begin{bmatrix} m_j I_3 & 0 \\ 0 & J_j(q) \end{bmatrix}
\]

with mass \( m_j \) and \((3 \times 3)\) inertia tensor \( J_j(q) \). \( v \) is the time derivative of the vector of generalized velocities \( \dot{v} \) with \( v_i = [v^T \omega^T]_i^T \). \( \lambda_{ext} \) is the vector of generalized forces acting on the bodies (gravity, velocity-product, and centripetal forces), \( \lambda \) is the vector of contact force magnitudes (one element for each contact), and \( G(q) \) is composed of the corresponding contact Jacobians where for the \( j^{th} \) contact

\[
G_{ij} = \begin{bmatrix} \hat{n}_{ij} \\ r_{ij} \times \hat{n}_{ij} \end{bmatrix}
\]

where \( r_{ij} \) is the vector from the center of mass of body \( i \) to the point of application of the force due to the \( j^{th} \) contact as in Figure 1 above. The generalized body configurations are represented by \( q = [x^T q^T] \) for position \( x \) and unit quaternion \( q \) [BETC12].

In addition to satisfying the Newton-Euler equation, rigid bodies should not interpenetrate. The standard approach to representing non-penetration constraints originated from the work of Signorini on problems of contact in solid mechanics [Sig59]. For the case of rigid body dynamics, the Signorini condition, also known as the “normal complementarity condition,” is written as

\[
0 \leq \psi_n(q, t) \perp \lambda_n \geq 0
\]

for a contact where \( \psi_n(q, t) \) is the signed gap distance which is dependent on the configuration of the bodies \( q \) and time \( t \), and \( \lambda_n \) is the magnitude of the force applied in the normal direction \( \hat{n} \) of the contact. Just as we will do for \( G \) and \( M \) below, we drop the functional dependence (\( \cdot \)) to simplify notation.

The orthogonality condition of equation (4) enforces for each contact that if the force being applied is positive (compressive) then the gap distance must be zero, and if the gap distance is positive then the force must be zero. The inequalities imply that the force cannot be tensile and the bodies cannot overlap. In the frictionless case, the continuous-time dynamic model of rigid bodies in contact is composed of equations (3) and (4).

**Discrete-Time Model**

Since no closed-form solution of the continuous-time model exists, a discrete-time model must be derived and solved numerically. One can discretize the Newton-Euler equation in several ways. For simplicity, we use a backward Euler approximation of the time derivatives over a time step \( h \) from step \( \ell \) to \( \ell + 1 \) to obtain

\[
Mv^{\ell+1} = M\dot{v}^{\ell} + G_n\rho_n^{\ell+1} + p_{ext}^{\ell}
\]

where \( p \) denotes an impulse over the time step, i.e., \( p = \lambda h \).

The gap function in general does not have a closed-form solution, but distances for various known configurations can be computed efficiently with collision detection algorithms. Therefore, the gap distance at the end of the next time step is represented as its Taylor series expansion, which we truncate after the linear term here (although truncation could be done after a high-order term if desired) to get

\[
\psi^{\ell+1} \approx \psi^{\ell} + G^T(q)\psi^{\ell+1}h + \frac{\partial \psi^{\ell}}{\partial t}h
\]

The non-penetration constraint of equation (4) is then discretized to

\[
0 \leq \psi_n^{\ell} + G_n^T\psi^{\ell+1}h + \frac{\partial \psi_n^{\ell}}{\partial t}h \perp \rho_n^{\ell+1} \geq 0
\]

Letting \( \rho_n^{\ell+1} = \frac{\psi_n^{\ell} + G_n^T\psi^{\ell+1} + \frac{\partial \psi_n^{\ell}}{\partial t}h}{h} \), we can write a dynamic model in the form of a mixed LCP (MCP) as

\[
\begin{bmatrix} 0 \\ \rho_n^{\ell+1} \end{bmatrix} = \begin{bmatrix} -M & G_n^T \\ G_n & 0 \end{bmatrix} \begin{bmatrix} \psi^{\ell+1} \\ \rho_n^{\ell+1} \end{bmatrix} + \begin{bmatrix} M\dot{v}^{\ell} + p_{ext}^{\ell} \\ \psi_n^{\ell} + \frac{\partial \psi_n^{\ell}}{\partial t} \end{bmatrix}
\]

\[
0 \leq \rho_n^{\ell+1} \perp \rho_n^{\ell+1} \geq 0
\]

Equations (8) and (9) define the time-stepping subproblem of Stewart and Trinkle [ST96] without friction. The first row of equation (8) is the discrete-time Newton-Euler equation. The second row is merely a definition used to shorten the statement of the normal complementarity condition (9). Importantly, equation (9) defines the standard model of unilateral contact used widely in multi-body simulators. The term \( \psi_n^{\ell} \), appearing in the standard model, acts as a constraint stabilization term, which if removed, allows interpenetrations to occur and then remain indefinitely. Regardless of whether term \( \psi_n^{\ell} \) is removed, or retained, the inequality on the left-hand side of condition (9) is the source of SM-traps, that we will see PEG eliminates.

**State Update**

Equations (8) and (9) models all bodies that were found to be in contact during the time step. Bodies that were not in contact are assigned updated velocities according to

\[
\dot{v}^{\ell+1} = \dot{v}^{\ell} + M^{-1}p_{ext}^{\ell}
\]
At the end of each time step, body configurations are updated. Given a quaternion representation of rotation of \( q = a + bi + cj + dk \), all dynamic bodies are then updated according to

\[
q^{t+1} = q^t + \left[ \begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & a & -c & b \\
0 & -c & a & b \\
0 & -b & -c & d \\
\end{array} \right] v^{t+1} h
\]  

Since equation (10) does not preserve length, each quaternion should be re-normalized after each time step.

1. A new approach

One approach to multibody simulation is to allow interpenetrations (and other constraint violations) to occur and then apply whatever hacks are necessary to catch poorly defined geometric configurations that inevitably result. Our approach attempts to prevent non-physical configurations from occurring. In order to achieve this we must detect potential contacts before interpenetration occurs.

Contact constraints have been traditionally modeled as fundamentally unilateral, however, representing features as infinite half-spaces at points of potential contact generates non-physical behaviour at corners where multiple finite features meet and terminate. Consider a vertex \( v \) near a corner as depicted in Figure 2. When the collision detection routine identifies both \( C_1 = C(v, e_1) \) and \( C_2 = C(v, e_2) \), the vertex becomes erroneously trapped in the convex subspace of the free space surrounding the corner if both contacts are enforced. We refer to this as the standard-model trap (SM-trap).

![Fig. 2: The standard-model trap. A vertex \( v \) approaching two edges \( e_1 \) and \( e_2 \) at a corner of a body \( B \). If unilateral constraints are enforced against both edges, the vertex becomes trapped by the sum of the edges’ half-spaces (light gray regions plus body). The same trap can occur in 3D between a vertex and set of faces or an edge against multiple edges.](image)

Figure 3 shows a simple situation in which two SM-traps collude to generate a time-stepping sub-problem with no solution. A common heuristic in collision detection is to choose contacts with deepest penetration. These contacts are then used to formulate a dynamics problem to move the simulation forward in time. Since the standard contact model requires that by the end of the next time step both of the identified contacts must be nonnegative, an impossible problem is generated. Poor contact choices lead to poorly defined dynamics problems, resulting in the all too familiar “popping” or “exploding” bodies in a simulation. Note that allowing bodies such as those in Figure 3 to first interpenetrate before identifying contacts does not avoid this erroneous contact problem, as the same heuristic results in similarly poorly defined dynamics. Fundamental errors such as these are what lead to common ad hoc “corrections” to these problems, such as limiting the maximum magnitudes of contact forces.

![Fig. 3: Infeasibility caused by the standard contact model. Vertex \( v_{a1} \) is penetrating the line containing edge \( e_{b3} \) (i.e., \( \psi_1 < 0 \)). Similarly, \( v_{a2} \) is penetrating the line containing edge \( e_{b1} \) (i.e., \( \psi_2 < 0 \)). Since the bodies are rigid, it is impossible to eliminate both penetrations at the end of the next time step. Therefore, the time-stepping sub-problem with constraint stabilization will not have a solution.](image)

**Deriving PEG, the new model**

Let us begin by solving the standard-model trap in 2D for a single vertex \( v \) approaching a corner composed of two edges \( e_1 \) and \( e_2 \). We identify two potential contacts \( C_1 = C(v, e_1) \) and \( C_2 = C(v, e_2) \) with corresponding gap distances \( \psi_1 \) and \( \psi_2 \). We wish to enforce that

\[
\max(\psi_1, \psi_2) \geq 0, 
\]  

allowing \( v \) to penetrate either half space corresponding to \( e_1 \) or \( e_2 \), but not both.

**Lemma 1.1.**

*Given \( a, b \in \mathbb{R}, \max(a, b) = a + \max(b - a, 0) \geq 0*

*Proof:*

*Case \( a > b \) \( \implies \) \( a + \max(b - a, 0) = a + 0 = a \)

*Case \( a \leq b \) \( \implies \) \( a + \max(b - a, 0) = a + (b - a) = b \) \( \square \)
Lemma 1.2.
Given $a, b \in \mathbb{R}, b = \max(a, 0) \iff 0 \leq b - a \perp b \geq 0$
Proof:
Case $a \leq 0$: $\max(a, 0) = 0 \implies b = 0 \implies 0 \leq b - a \perp b \geq 0$
Case $a > 0$: $\max(a, 0) = a \implies b = a \implies 0 \leq b - a \perp b \geq 0$
Case $b - a = 0, b \geq 0$: $b = \max(a, 0) = a$
Case $b = 0, b - a \geq 0$: $a \leq 0 \implies \max(a, 0) = 0 = b \quad \square$

Using Lemma 1.1, we rewrite and constrain equation (11) as

$$\max(\psi_1, \psi_2) = \psi_1 + \max(\psi_2 - \psi_1, 0) \geq 0$$

Then using Lemma 1.2 and letting $c = \max(\psi_2 - \psi_1, 0)$, we can constrain $\max(\psi_1, \psi_2) \geq 0$ with

$$0 \leq c + \psi_1 - \psi_2 \perp c \geq 0 \quad \geq 0$$

(12)

To give some insight into the slack variable $c$ and its relationship with $\psi_1$ and $\psi_2$, consider the figure below and the value of $c$ on each side of the bisector where $\psi_1 = \psi_2$. A vertex $v$ in the free space right of the bisector has $\psi_1 > \psi_2 \implies \psi_1 - \psi_2 > 0$ which requires $c = 0$ to satisfy the complementarity condition of (12). A vertex to the left of the bisector has $\psi_1 < \psi_2 \implies \psi_1 - \psi_2 < 0$ which requires $c = \psi_2 - \psi_1$.

We now wish to include potential forces $\lambda_1$ and $\lambda_2$ while ensuring that only one is non-zero since allowing both to be simultaneously non-zero would be non-physical.

Lemma 1.3.
Given $a, b \in \mathbb{R}, b = |\min(a, 0)| \iff 0 \leq b + a \perp b \geq 0$
Proof:
Case $a \leq 0$: $|\min(a, 0)| = -a \implies b + a = 0 \implies 0 \leq b + a \perp b \geq 0$
Case $a > 0$: $|\min(a, 0)| = 0 \implies (a + b)b = 0 \implies 0 \leq b + a \perp b \geq 0$
Case $b + a = 0, b \geq 0$: $a \leq 0 \implies |\min(a, 0)| = -a = b$
Case $b = 0, b + a \geq 0$: $a > 0 \implies |\min(a, 0)| = 0 = b \quad \square$

Lemma 1.4.
Given $a, b \in \mathbb{R}, a = 0, b \leq 0 \iff (\max(a, b) \geq 0) \land (\max(a, b) + |\min(a, 0)| = 0)$

Proof:
The only way to satisfy both $\max(a, b) \geq 0$ and $|\min(a, 0)| \geq 0$ is for both to reach equality. When $|\min(a, 0)| = 0$, it is possible that $a \geq 0$, so in order to have $\max(a, b) = 0$ we must have $a = 0, b \leq 0$. The argument for the reverse of $a$ and $b$ is similar. \square

Using Lemma 1.3, we write

$$0 \leq d_1 + \psi_1 \perp d_1 \geq 0$$

and using Lemma 1.4, we have

$$0 \leq c + \psi_1 + d_1 \perp \lambda_1 \geq 0$$

Essentially, we have taken a set of contacts, in this case $C_1$ and $C_2$, and replaced a logical AND condition of $C_1 \land C_2$ with a NAND condition of $C_1 \bar{\land} C_2$. Geometrically, this corresponds to replacing line representations of a body with line segments. At this point, equations (12), (13), and (14) correspond to the locally non-convex (LNC) model proposed by Nguyen [Ngu11] which models the case for a single vertex against multiple edges. Although this model has been shown to contribute some improvement in physical fidelity over other methods [FT13], stopping here would correct the standard-model trap at the expense of permitting interpenetrations [Wil14a, WLT14]. To illustrate this error, let us continue with a less trivial case.

Consider two bodies approaching as depicted in Figure 5. In this configuration, our collision detection routine identifies four potential contacts. These are $C_1 = \mathcal{C}(v_a, e_{b1})$, $C_2 = \mathcal{C}(v_a, e_{b2})$, $C_3 = \mathcal{C}(v_b, e_{a1})$, and $C_4 = \mathcal{C}(v_b, e_{a2})$. We should not simply constrain all of these contacts, for this would result in a dual standard-model trap. It is possible to make a guess as to which contacts to enforce, but this inserts the likely possibility of generating contact forces where...
there are none. Of course we could also ignore all four contacts and allow interpenetration, but this too is clearly a non-physical approach and leads to instabilities.

Instead, let us write constraints including all four potential contacts in such a way as to allow for all physically feasible interactions while preventing interpenetration. We start by observing that constraints for \( C_1 \) and \( C_2 \) should both be able to be violated, but not simultaneously. This is similarly true for \( C_3 \) and \( C_4 \). We have seen how these constraints can be represented with

\[
0 \leq c_1 + \psi_1 - \psi_2 \perp c_1 \geq 0 \\
0 \leq c_2 + \psi_4 - \psi_3 \perp c_2 \geq 0 \\
        c_1 + \psi_1 \geq 0 \\
        c_2 + \psi_4 \geq 0 \\
        0 \leq d_1 + \psi_1 \perp d_1 \geq 0 \\
        0 \leq d_2 + \psi_2 \perp d_2 \geq 0 \\
        0 \leq d_3 + \psi_3 \perp d_3 \geq 0 \\
        0 \leq d_4 + \psi_4 \perp d_4 \geq 0 \\
0 \leq c_1 + \psi_1 + d_1 \perp \lambda_1 \geq 0 \\
0 \leq c_1 + \psi_2 + d_2 \perp \lambda_2 \geq 0 \\
0 \leq c_2 + \psi_3 + d_3 \perp \lambda_3 \geq 0 \\
0 \leq c_2 + \psi_4 + d_4 \perp \lambda_4 \geq 0 \\
\]

Equation (15) can be completed by including constraints to prevent such inter-penetrations. That is, we additionally constrain \( \max(\psi_2, \psi_3) \geq 0 \) with

\[
0 \leq c_3 + \psi_1 - \psi_4 \perp c_3 \geq 0 \\
0 \leq c_4 + \psi_2 - \psi_3 \perp c_4 \geq 0 \\
        c_3 + \psi_1 \geq 0 \\
        c_4 + \psi_2 \geq 0 \\
\]

Equations (15) and (16) together represent our first example of a geometrically accurate contact model. We will refer to the constraint type in equation (15) as \( I \)-constraint, and the type in equation (16) as \( C \)-constraint. These, along with unilateral constraint, form the set of three contact constraints necessary for our model.

2 Three fundamental contact constraints

In order to resolve the issues discussed in the previous section, we had to change the way we formulate contact constraints. In this section, we define three fundamental contact constraints in their general form and begin to describe how they can be applied.

2.1 Unilateral contact constraint

The simplest of the three contact constraints is the common unilateral contact constraint (\( U \)-constraint) that was introduced by equation (4). We define a \( U \)-constraint as a function of a single contact \( C_i \) and write it as

\[
U(C_i) \\
\]

which corresponds to the complementarity problem

\[
0 \leq \psi_i \perp \lambda_i \geq 0 \\
\]

where \( \lambda_i \) is the force corresponding to contact \( C_i \).

2.2 Inter-contact constraint

An inter-contact constraint (\( I \)-constraint) is that which was demonstrated in equation (15) for preventing the standard-model trap. An \( I \)-constraint deals with contacts of a single feature of one body against multiple features of another body. Here we will define the general formulation of \( I \)-constraint for a vertex of \( A \) against multiple features of \( B \) where a subset of these potential contacts may feasibly result in a contact force at the end of the current time step.

Consider two sets of contacts: \( C_i \) which is composed of all contacts which could feasibly result in a contact force, and \( C_k \) which contains all contacts that could not feasibly result in a contact force but are necessary to accurately represent the contact geometry. An \( I \)-constraint seeks to

- Constrain at least one gap distance \( \psi_{i,k} \) of \( \{C_i, C_k\} \) to be non-negative,
- Permit a non-zero contact force \( \lambda_{i,i} \) for at most one contact in \( C_i \), and
- Only permit \( \lambda_{i,i} > 0 \) when \( \psi_{i,i} = 0 \) and all other \( \psi_{i,k} \) are non-positive.

We represent an \( I \)-constraint as

\[
I(C_i, C_k) \\
\]

which corresponds for \( m = |C_i| \) and \( n = |C_k| \) to the set of
constraints

\[ 0 \leq c_{12} + \psi_{t1} - \psi_{t2} \perp c_{12} \geq 0 \]
\[ 0 \leq c_{13} + c_{12} + \psi_{t1} - \psi_{t3} \perp c_{13} \geq 0 \]
\[ \vdots \]
\[ 0 \leq c_{1m} + c_{1m-1} + \ldots + c_{12} + \psi_{t1} - \psi_{tm} \perp c_{1m} \geq 0 \]
\[ 0 \leq c_{k1} + c_{1m} + c_{1m-1} + \ldots + c_{12} + \psi_{t1} - \psi_{k1} \perp c_{k1} \geq 0 \]
\[ \vdots \]
\[ 0 \leq c_{km} + \ldots + c_{k1} + c_{1m} + \ldots + c_{12} + \psi_{t1} - \psi_{km} \perp c_{km} \geq 0 \]  

Equations (20) and (21) enforce that at least one contact in \( \{ C_i, C_k \} \) is non-negative. Equations (22) and (23) allow at most one contact force to be non-zero, and only permit this force to be non-zero when all other gap distances are non-positive. Solving such a set of constraints is possible as a linear program with complementarity constraints (LPCC) [HMP+08], or by introducing an auxiliary variable to convert inequalities (21) into a linear complementarity conditions before using an LCP solver [Ngu11].

An \( \mathcal{I} \)-constraint requires there be at least one contact in \( C_i \). If there were not, then there would exist no non-trivial solution since these potential contact forces are necessary for preventing interpenetration. In the case that \( |C_i| = 1 \), we are able to dramatically reduce the set of constraints by eliminating equations (21) and (22) and replacing equation (23) with the single constraint

\[ 0 \leq c_{km} + \ldots + c_{k1} + c_{1m} + \ldots + c_{12} + \psi_{t1} + d_{11} \perp \lambda_{k1} \geq 0 \]
\[ \vdots \]
\[ 0 \leq c_{km} + \ldots + c_{k1} + c_{1m} + \ldots + c_{12} + \psi_{t1} + d_{lm} \perp \lambda_{lm} \geq 0 \]  

This greatly reduces the number of constraints, and removes the inequality of (21) to leave a pure LCP. We see now the incentive for choosing our contact set with a single primary contact \( C_i \), possibly using heuristics like those described in section 3.

### 2.3 Cross-contact constraint

In contrast to \( \mathcal{I} \)-constraint which deals with a single feature of \( A \) against multiple features of \( B \), a cross-contact constraint (\( \mathcal{X} \)-constraint) deals with sets of contacts that may not depend on the same features, but could result in interpenetration if simultaneously violated, e.g., equation (16).

Given sets of contacts \( C_a \) and \( C_b \), a \( \mathcal{X} \)-constraint attempts to constrain either

- At least one non-negative \( \psi_{ka} \) in \( C_a \)
  - OR
- At least one non-negative \( \psi_{kb} \) in \( C_b \)

We represent a \( \mathcal{X} \)-constraint as

\[ \mathcal{X}(C_a, C_b) \]

and can write the LCP in multiple forms. Conveniently, \( \mathcal{X} \)-constraints can be written in the form of equations (20-21). That is, \( \mathcal{X}(C_a, C_b) \) is similar to \( \mathcal{I}(C_a, C_b) \), but excludes equations (22-23) which involve contact forces.

Although a \( \mathcal{X} \)-constraint necessarily deals with contacts between the same pair of bodies, the contacts need not share features. That is, the geometric interdependencies of identified potential contacts, regarding which are to be enforced as defined by an \( \mathcal{X} \)-constraint, may be between two otherwise seemingly independent contacts, e.g., opposite corners of the face of a box. Subsequently, it is conceivable for two potential contacts neither of which have a potential force associated with them, i.e., contacts in two different sets of \( C_k \), to be \( \mathcal{X} \)-constrained. In such cases, the dynamics problem requires sufficient \( \mathcal{U} \)- or \( \mathcal{I} \)-constraints so as to allow generation of forces which may contribute to achieving configurations that will satisfy all constraints. This issue however, while an important one when employing a heuristic approach to choosing primary contacts in \( \mathcal{I} \)-constraints, is beyond the scope of this paper.

### 3 Contact between moving geometries

In this section, we briefly discuss the nature of interpolytope contact, and define a set of geometric tests that are useful for identifying and making heuristic choices regarding potential contacts.

We define a convex polyhedron in terms of features. We may refer to a vertex either as an object \( v \) or as the vector

![Fig. 7: Geometric representation of a convex polyhedron.](image)
that are planar with \( \hat{n}_f \) for a given face \( f \in F \) is perpendicular to \( f \). We define two vectors \( t_1 \) and \( t_2 \) for each edge that are planar with \( f_1 \) and \( f_2 \) respectively, and perpendicular to \( e \). These vectors are defined as

\[
\begin{align*}
    t_1 &= \hat{n}_1 \times e \\
    t_2 &= e \times \hat{n}_2
\end{align*}
\]  

(26)

and are useful for geometric tests when determining contacts. We additionally define a convention of edge orientation \( O_e \) as

\[
O_e = (\hat{n}_1 + \hat{n}_2) \times e.
\]

(27)

for convex bodies.

As we develop these tests, we keep in mind that a contact can occur anywhere on a polytope, but the possible normal vectors are restricted by the contacted feature. Figures 8 through 12 depict the possible positions and orientations of contact normal vectors for each feature in 2 and 3 dimensions. Since a vertex does not have a uniquely defined normal, we use the normal from the edges that connected to the vertex. It is important to realize that when using a time-stepping method, we hope to identify contacts which will reflect these feasible contact normals at the end of the time step. Yet, many methods of contact identification use approaches that assume continuous or static body positions. This is why we geometrically relax the following set of tests that will prove useful for identifying and making heuristic choices regarding potential contacts.

Consider the simple example in Figure 13 where two vertices, \( v_{a1} \) and \( v_{a2} \) of body \( A \) are near an edge \( e_b \) of body \( B \). In order to achieve stability in such a configuration, it is clear that both potential contacts \( C_1 = C(v_{a1}, e_b) \) and \( C_2 = C(v_{a2}, e_b) \) must be considered by the dynamics formulation. If only one were included, \( e_b \), then it is possible for the other to penetrate \( e_b \) given a large enough time step or velocity. We observe that the second contact normal \( n_2 \) is outside of the normal cone region of \( v_{a2} \), does not correspond to a physically reasonable contact normal. However, this is because \( n_2 \) is determined at the beginning of the time step, yet we wish to prevent interpenetration for the end of the time step. In order to avoid interpenetrations, we must detect and consider such contacts as \( C_2 \), even when they are not physically feasible at the current time step. This necessity requires that we geometrically relax the normal cone region, or equivalently, relax the definition of applicability.

### 3.1 Contact applicability

The time required to solve a time-stepping subproblem, and therefore the time required to compute a simulation, is strongly dependent on the number of contacts \( n_c \). While it may seem that the number of contacts is determined by the simulation, taking a close look at an individual discrete time step, as we do in section “Discrete-Time Model”, reveals that to enhance simulation accuracy and reduce instability, it is
beneficial to include “potential contacts” even though they increase the problem size. We slightly modify classical applicability conditions to derive a method to choose a small set of extra contacts.

The geometric test of applicability is well known [Don84, Lat91, Lin93, JTT98] and commonly used in applications such as distance computation and collision detection for motion planning with polytopal objects. We geometrically relax the traditional motion of applicability in order to better determine which contacts to include. We have adopted notation similar to that of Latombe [Lat91]. The basic idea is shown in Figure 13. If

\[ \text{APPL}^{(\text{ve})}(q, v_a, e_b) \geq \epsilon_\theta \text{ for relaxation parameter } \epsilon_\theta = -\sin(\theta) \]

and where

\[ \text{APPL}^{(\text{ve})}(q, v_a, e_b) = \min_{v_k \in \omega(v_a)} \hat{n}_b \cdot \frac{v_k - v_a}{||v_k - v_a||} \] (29)

Whereas classical applicability is a boolean function, our function returns a value as we may wish to use the value of applicability as a heuristic when comparing contacts.

3D applicability

The 3D vertex-face case is analogous to the 2D vertex-edge case and is defined as

\[ \text{APPL}^{(\text{vf})}(q, v_a, f_b) = \min_{v_k \in \omega(v_a)} \hat{n}_b \cdot \frac{v_k - v_a}{||v_k - v_a||} \] (30)

where \( \hat{n}_b \) is the outward unit normal of \( e_b \), and the superscript \( \text{vf} \) denotes that the test is for a vertex and a face.

Notice that the applicability test is dependent on \( q \), more specifically, it is dependent on the relative orientation of the polygons, and not their relative translation. Later, when considering potential contacts to include in a time-stepping subproblem, we will introduce a relaxed form of inequality (28) to allow for possible rotation in a time step. In the case of our example, if the red polygon rotated enough in the clockwise direction in the time-step under consideration, contact could occur first at vertex \( v_{a2} \) rather than vertex \( v_{a1} \).

2D applicability

Let \( \omega(v) \) be the set of all vertices connected by an edge to \( v \). We refer to \( v_a \) having applicability with \( e_b \) when

\[ \text{APPL}^{(\text{ve})}(q, v_a, e_b) = \bigwedge_{v_k \in \omega(v_a)} \hat{n}_b \cdot \frac{v_k - v_a}{||v_k - v_a||} \geq 0 \] (28)

where \( \hat{n}_b \) is the outward unit normal of \( e_b \), and the superscript \( \text{ve} \) denotes that the test is for a vertex and an edge.

Figure 15 depicts two edges, \( e_a \) and \( e_b \), in contact. Just as there are vertex-face configurations that have proximity but could not result in a force, so too are there edge-edge configurations that should not be associated with potential contact forces. First, we should note that given an edge-edge contact normal \( n = e_a \times e_b \), we cannot immediately say if we have the correct sign for \( n \). However, we can still determine edge-edge applicability in the following manner. Let

\[ d_{a1} = \hat{n} \cdot \hat{t}_{a1} \]
\[ d_{a2} = \hat{n} \cdot \hat{t}_{a2} \]
\[ d_{b1} = -\hat{n} \cdot \hat{t}_{b1} \]
\[ d_{b2} = -\hat{n} \cdot \hat{t}_{b2} \]

We then define edge-edge applicability as

\[ \max(\min(d_{a1}, d_{a2}, d_{b1}, d_{b2}), \min(-d_{a1}, -d_{a2}, -d_{b1}, -d_{b2})) \] (31)
Fig. 15: Edge-edge contact at the border of stability for classical applicability ($\varepsilon^{ee}_a = 0$). The result in this configuration is dependent on machine precision error. Relaxation eliminates this dependency by increasing the domain of applicability.

Conceptually similar to (30), (31) measures the maximal extent by which the adjacent faces of an edge are pointed “in” on another body.

3.2 Contact feasibility

Feasibility is similar to applicability, but is dependent on feature positions as opposed to orientations. The region of feasibility is essentially the region in which we anticipate a contact between a vertex $v$ and a facet $f$ could occur. We define vertex-edge and vertex-face feasibility similarly. Given an angle $\phi$ and a small value $\delta$ greater than machine precision, a vertex with contact $C(v_a, f_b)$ has feasibility if

- $v_a \in VR(f)$ and $\psi \geq -\delta$
- OR
- $v_a \notin VR(f)$ and $v_a$ within region defined by $\phi$ below the point $\varepsilon$ below the nearest point on $f$.

where $VR(f)$ refers to the Voronoi region of the facet $f$ [Mir98]. An example of this region is depicted in Figure 16.

We can also write an edge-edge feasibility in 3D, which checks only to see if the nearest two points $p_a$ on $e_a$ and $p_b$ on $e_b$ are within some small distance $\varepsilon$.

4 Polytopal Exact Geometry

From the set of fundamental constraints presented in section 2, which are written in terms of complementarity problems, abstract out another step and look at pairs of bodies by cases of feature pairs.

This abstraction layering is depicted in Figure 17. At the lowest level is the complementarity problem which is utilized first as a model of unilateral contact with force, but further as a tool for generating logical NAND conditions on subsets of potential contacts. Sets of these low level constraints can be entirely represented by abstracting to the three fundamental constraints of unilateral, inter-contact, and cross-contact. In turn, the fundamental constraints are formulaically generated by the highest abstraction layer which looks only at what pairs of features between two bodies are near each other. In the next section, we will see how precisely the same abstraction can be developed in 3D.

4.1 PEG 2D

We consider two contact configurations in 2D: vertex-edge and vertex-vertex. We do not consider edge-edge because this case is redundantly covered by the other two configurations [Lat91]. To be clear, edge-edge penetration in 2D is impossible if vertex-edge and vertex-vertex constraints are properly enforced, simply because edges defined and composed of vertices, and for two edges to come near each other implies that the vertices of at least one of those edges is nearing the other edge. We can formulaically generate the set of constraints for the two 2D configurations as given below.

Vertex-edge 2D

Given a vertex $v_a$ near a single edge $e_b$, this case is modeled by a unilateral constraint on a single contact as

$$U(C(v_a, e_b))$$

(32)
In this case, the vertex is within a reasonably small distance $\varepsilon$ of the edge, but farther than $\varepsilon$ from either of the edge's vertices.

**Vertex-vertex 2D**

The 2D vertex-vertex case was depicted in Figure 5. Using the contacts defined for this case requires the constraints

$$
\begin{align*}
\mathcal{I}(C_{11}, C_{k1}) \\
\mathcal{I}(C_{12}, C_{k2}) \\
\mathcal{X}([\{C_1\}, \{C_4\}]) \\ 
\mathcal{X}([\{C_2\}, \{C_3\}])
\end{align*}
$$

where $C_{11}$ contains at least one of $C_1$ or $C_2$, $C_{k1}$ may contain one of $C_1$ or $C_2$, and similarly $C_{12}$ contains at least one of $C_3$ or $C_4$, and $C_{k2}$ may contain one of $C_3$ or $C_4$. As is the case for all $\mathcal{I}$-constraints, which contacts are included in $C_1$ or $C_2$ are determined using the geometric heuristics defined in section 3.

4.2 PEG 3D

There are four contact configurations in 3D: vertex-face, edge-edge, vertex-edge, and vertex-vertex. Again, we exclude certain feature pairs just as we did in 2D, namely edge-face and face-face, because they become redundant when the other configurations are properly enforced. Edge-face is redundant since edges are composed of vertices, and for an edge to approach a face requires that either the vertices of that edge approach that face, or the edge approaches the edges at the perimeter of that face (possibly adjacent edges), reducing to either vertex-face or edge-edge respectively, and edge-vertex in the case that the edges of the second body are adjacent. By a similar argument, face-face becomes redundant; a face, being composed of edges in turn composed of vertices, nearing another face implies that edges and vertices are approaching the other face. In all possible configurations, this results in one of the four feature combinations considered.

**Vertex-face 3D**

The vertex-face case is analogous to the 2D vertex-edge case. Given a vertex $v_a$ near a face $f_b$ as depicted in Figure 14, we include

$$
\mathcal{U}(C(v_a, f_b))
$$

in the set of constraints.

**Edge-edge 3D**

Edge-edge contact, such as depicted in Figure 15, is another example of a single contact with unilateral constraint, and is written for an edge $e_a$ against an edge $e_b$ by

$$
\mathcal{U}(C(e_a, e_b))
$$

**Vertex-edge 3D**

Consider Figure 18 where a vertex $v_a$ of body $A$ approaches the edge $e_b$ of body $B$. There are an arbitrary number of edges (greater or equal to three) that could connect to $v_a$, however we can limit the edge-edge contacts considered to a subset of those with relatively large magnitudes of edge orientation $|e_a - O_{e_b}|$. Given $C_1 = C(v_a, f_1)$ and $C_2 = C(v_a, f_2)$, the set of constraints for vertex-edge is given by

$$
\begin{align*}
\mathcal{I}(C_1, C_2) \\
\mathcal{X}([\{C_{ei}\}, \{C_1\}]) \\
\mathcal{X}([\{C_{ej}\}, \{C_2\}])
\end{align*}
$$

where $C_1$ and $C_2$ are distributed appropriately into $C_{ei}$ and $C_{ej}$. Each $C_{ei}$ is an edge with positive edge orientation, and each $C_{ej}$ is an edge with negative edge orientation, and both sets of edges have applicability and feasibility with $e_b$. In the case of a vertex $v_a$ near another body’s vertex $v_b$ as depicted in Figure 19, we wish to avoid trapping $v_a$ against the faces of $B$, and similarly for $v_b$ against $A$. This first requires the set of $\mathcal{I}$-constraints

$$
\begin{align*}
\mathcal{I}(C_{ba}, C_{ka}) \\
\mathcal{I}(C_{tb}, C_{kb})
\end{align*}
$$

where $C_{ba}$ and $C_{ka}$ are composed of contacts between $v_a$ and faces of $B$, and $C_{tb}$ and $C_{kb}$ are composed of contacts between $v_b$ and faces of $A$. We additionally require $\mathcal{X}$-constraints for each face pair $(f_a, f_b)$ in the form

$$
\mathcal{X}([C(v_a, f_b), C(v_b, f_a)], [C(e_{a1}, e_{b1}), C(e_{a2}, e_{b2})])
$$
where $f_a$ has edges $e_{a1}, e_{a2}$ connected to $v_a$, and $f_b$ has edges $e_{b1}, e_{b2}$ connected to $v_b$. These $X$-constraints are enforced only for edges $e_{a1}$ and $e_{b1}$ which have edge-edge applicability and feasibility. The constraint in equation (38) is similar to the $X$-constraints in equation (36) for the vertex-edge case, but permits a vertex to pass on either side of a face instead of either side of an edge.

![Fig. 19: A vertex $v_a$ of body $A$ approaching a vertex $v_b$ of body $B$ from above. Vertex $v_a$ should be permitted to break the half spaces represented by the falsely extended faces of $B$, but not all of them and only when doing so does not violate edge-edge contacts.](image)

For $m$ $I$-constraints, $c_I$ is the solution vector composed of unknown slack variables

$$
c_I^{\ell+1} = 
\begin{bmatrix}
c_{I_1}^{\ell+1} \\
\vdots \\
c_{I_m}^{\ell+1}
\end{bmatrix}
$$

where the cardinality of each $c_{I_j}^{\ell+1}$ is $k$ and the $j^{th}$ $I$-constraint has $k + 1$ potential contacts, $E_{I_{1j}}$ is diagonally composed of sub-matrices where

$$
E_{I_{1j}} =
\begin{bmatrix}
1 & 0 & 0 \\
\vdots \\
1 & \cdots & 0 \\
1 & 1 & 1
\end{bmatrix}
$$

which is the lower triangular matrix of size $k$, $E_{I_{2j}}$ is diagonally composed of sub-matrices of row vectors of ones where

$$
E_{I_{2j}} = [1 \ 1 \ \cdots \ 1]
$$

has length $k$, $\Psi_I$ we will define as

$$
\Psi_I = 
\begin{bmatrix}
\Psi_{I_1} \\
\Psi_{I_{1j}} \\
\vdots \\
\Psi_{I_m}
\end{bmatrix}
$$

where each sub-vector $\Psi_{I_j}$ is given by

$$
\Psi_{I_j} =
\begin{bmatrix}
\frac{\psi_{I_{1j}}}{h} + \frac{\partial \psi_{I_{1j}}}{\partial t} \\
\vdots \\
\frac{\psi_{I_{kj}}}{h} + \frac{\partial \psi_{I_{kj}}}{\partial t}
\end{bmatrix}
$$

$\Psi_{I_1}$ is the vector of primary potential contacts

$$
\Psi_{I_1} = 
\begin{bmatrix}
\frac{\psi_{I_{11}}}{h} + \frac{\partial \psi_{I_{11}}}{\partial t} \\
\vdots \\
\frac{\psi_{I_{1m}}}{h} + \frac{\partial \psi_{I_{1m}}}{\partial t}
\end{bmatrix}
$$

$\mathbf{p}_I^{\ell+1}$ is a vector of unknown impulses

$$
\mathbf{p}_I^{\ell+1} = 
\begin{bmatrix}
\mathbf{p}_{I_1}^{\ell+1} \\
\vdots \\
\mathbf{p}_{I_{1j}}^{\ell+1} \\
\vdots \\
\mathbf{p}_{I_m}^{\ell+1}
\end{bmatrix}
$$

for which we also extend the Newton-Euler equations to include

$$
\mathbf{Mv}^{\ell+1} = \mathbf{M}v^{\ell} + \cdots + \mathbf{G}_{nI}\mathbf{p}_{nI}^{\ell+1}
$$

4.3 Time-stepping formulation with PEG

As an independent contact model, there are many methods into which we may incorporate PEG. Here however, we will use equation (6) to discretize the PEG contact constraints in the same manner as the Stewart-Trinkle method of equations (8) and (9). Additionally, we will focus on the case where we make a heuristic choice of a single primary contact, i.e., $|C_1| = 1$ for every set of dependent potential contacts. We have already seen the discretized unilateral constraint in equation (7). An $I$-constraint may be written in terms of velocity by substituting equation (6) into (20) and (24) to get

$$
0 \leq E_{I_{1j}} c_I^{\ell+1} + \frac{\psi_{I_1}}{h} + \mathbf{G}_{I_1}^T v^{\ell+1} + \frac{\partial \psi_{I_1}}{\partial t} \perp c_I^{\ell+1} \geq 0
$$

and

$$
0 \leq E_{I_{2j}} c_I^{\ell+1} + \frac{\psi_{I_1}}{h} + \mathbf{G}_{I_1}^T v^{\ell+1} + \frac{\partial \psi_{I_1}}{\partial t} \perp \mathbf{p}_I^{\ell+1} \geq 0
$$

for which we also extend the Newton-Euler equations to include

$$
\mathbf{Mv}^{\ell+1} = \mathbf{Mv}^{\ell} + \cdots + \mathbf{G}_{nI}\mathbf{p}_{nI}^{\ell+1}
$$
\(G_I\) is the Jacobian difference matrix diagonally composed of sub-matrices \(G_{IJ}\) each composed of sub-matrices \(G_{Ijs}\) for \(s\) from 2 to \(k\) where

\[
G_{Ijs} = \left[ \hat{n}_{j1} \right] - \left[ \hat{n}_{js} \right] = \left[ \hat{n}_{js} \times \hat{n}_{j1} \right]
\]

(42)

and \(G_{nI}\) is the Jacobian matrix of only primary contacts

\[
G_{nI} = \left[ \hat{n}_{j1} \right] - \left[ \hat{n}_{j1} \times \hat{n}_{j1} \right].
\]

A \(X\)-constraint can be written in terms of velocities and impulses and incorporated in precisely the same way as the \(I\)-constraint. The result is that the PEG model may be written (without friction) as

\[
Mv^{t+1} = Mv + G_n\rho_n^{t+1} + G_{nI}\psi^{t+1} + G_{nX}\xi^{t+1} + \rho_{ext}
\]

(43)

\[
0 \leq \frac{\Psi_n}{h} + G_n^T\psi^{t+1} + \frac{\partial \Psi_n}{\partial t} \perp \rho_n^{t+1} \geq 0
\]

(44)

\[
0 \leq E_Ic_I^{t+1} + \frac{\Psi_I}{h} + G_I^T\psi^{t+1} + \frac{\partial \Psi_I}{\partial t} \perp c_I^{t+1} \geq 0
\]

(45)

\[
0 \leq E_Xc_X^{t+1} + \frac{\Psi_X}{h} + G_X^T\psi^{t+1} + \frac{\partial \Psi_X}{\partial t} \perp c_X^{t+1} \geq 0
\]

(46)

Considering that \(X\)-constraints may be represented mathematically as \(I\)-constraints with no impulses, we may simplify to

\[
\rho_n^{t+1} = G_n^T\psi^{t+1} + \frac{\Psi_n}{h} + \frac{\partial \Psi_n}{\partial t}
\]

\[
\rho_{nI}^{t+1} = G_{nI}^T\psi^{t+1} + E_Ic_I^{t+1} + \Psi_I
\]

\[
\gamma_I^{t+1} = G_I^T\psi^{t+1} + E_Xc_X^{t+1} + \Psi_I
\]

which may then be written in the form of a mixed LCP as

\[
\begin{bmatrix}
0 \\
\rho_n^{t+1} \\
\rho_{nI}^{t+1} \\
\gamma_I^{t+1}
\end{bmatrix} = 
\begin{bmatrix}
-M & G_n & 0 & 0 \\
G_n & 0 & 0 & 0 \\
G_{nI} & 0 & E_I & 0 \\
G_I & 0 & 0 & E_X
\end{bmatrix}
\begin{bmatrix}
\psi^{t+1} \\
\rho_n^{t+1} \\
\rho_{nI}^{t+1} \\
\gamma_I^{t+1}
\end{bmatrix} + 
\begin{bmatrix}
\psi^{t+1} \\
\rho_n^{t+1} \\
\rho_{nI}^{t+1} \\
\gamma_I^{t+1}
\end{bmatrix} = 
\begin{bmatrix}
Mv^{t} + \rho_{ext} \\
G_n^T + \rho_{ext} \\
G_{nI} + \rho_{ext} \\
G_I + \rho_{ext}
\end{bmatrix}
\]

(48)

Equations (48) and (49) constitute one possible time-stepping sub-problem that incorporates PEG for accurate geometric representations of angular features of interacting rigid frictionless polytopes. As mentioned above, friction may be added easily following the discussion in [BET14, Wil14b].

5 Results

The experiments discussed below compare simulations performed with the time-stepping methods defined by equations (8) and (9) and equations (48) and (49) in 2D and 3D. The only difference in the time steppers was that one used the standard contact model and the other used PEG. Both methods retained the term \(\Psi_n/h\) for constraint stabilization and time-stepping sub-problems were solved by PATH [FM00]. Other solvers were used, but only the results using PATH are provided here, because it performed better than the others on simulations with the standard model. For simulations with PEG, a single primary potential contact was chosen for each set of potential contact, which yielded the least accurate form of PEG. Despite this, the results show significant accuracy improvements when using PEG. Finally, to highlight the fact that PEG can easily be used with nonconvex bodies, the examples are of convex polytopes poured into nonconvex containers. The containers are represented as unions of convex polytopes, which required us to do a bit more bookkeeping, but the basic PEG algorithms did not need to be altered.

5.1 Box of polygons

Figure 20 shows a snapshot from the 2D experiment, which consisted of ten simulations using PEG and the same ten using the standard contact model. In each simulation, 20 randomly generated convex polygons were “poured” into an open box fixed in space. One polygon was dropped every 0.25s until the simulation ended at 5.0s. At the end of each time step, the area of overlap among all polygons was computed and used in two ways; first as an accuracy metric and second, for constraint stabilization. For the latter, each overlap was used to generate an effective contact point, contact normal, and penetration depth, and each was incorporated into the time-stepper as a \(I\)-constraint. When simulating the standard model, the simulation always crashed prior to reaching five seconds of simulation time, due to SM-traps that caused the solver to fail. Usually prior to failure, PATH returned solutions that allowed interpenetration and nonphysically large body velocities. Pivoting solvers, such as Lemke’s algorithm [CPS92], demonstrated similar difficulties finding physically sensible solutions in SM-trap situations. By contrast, time-stepping with PEG never encountered such problems.

Figure 21 shows the median and the first and third quartiles of area of overlap error for increasing time step size.
Fig. 20: A snapshot from a polygon pouring simulation. A polygon was dropped into a box every 0.25 seconds for 5 seconds of simulation. At each time step, the total area of overlap between all pairs of bodies was used as the performance metric.

Even for small time steps, the time-stepper using PEG produced errors several orders of magnitudes smaller than the time-stepper using the standard contact model.

5.2 Box of polyhedra
As shown in Figure 23, we extend the polygon pouring experiment to 3D. In the 3D simulations, the random polygons were replaced by random convex polyhedra, the box became the union of five right hexahedra, and volumes of overlap replaced the areas used in the 2D simulations. Again, for each overlap, the volume was computed, and from that, an effective contact point, contact normal, and penetration depth were computed to define a U-constraint to include in the active set. Because of the longer simulation times, fewer polyhedra were used. One was dropped into the box every 0.5 seconds until 5.0s of simulated time was reached.

The constraint violation error defined as volume overlap is depicted in Figure 24, which has extremely similar behaviour with the 2D results. As the step size increases, the penetrations between polyhedra grow steadily larger with the standard contact model, whereas PEG virtually eliminates this error at all time steps.

The occurrence of SM-traps is shown in Figure 25, where we see a rate of occurrence similar to the 2D experiment. Although the shape similarity is intriguing, it is simply coincidence, as these results are only representative of a single experiment and time step size for both 2D and 3D.

6 Conclusions and future work
There have been previous good results regarding physical fidelity and stability of differential variational inequality (DVI) methods [Ber09, Zha13, Mac12] upon which we have built our contact model. PEG further improves stability by avoiding interpenetrations, which can generate nonphysical behavior in ways that are difficult to predict and correct. Although PEG results in significantly smaller error while maintaining dynamic stability, there is still some increase in error.
Fig. 23: Sample frame from a polyhedron pouring simulation. A polyhedron was dropped into a container every 0.5 seconds for 5 seconds of simulation. At each time step, the total volume of overlap between all pairs of bodies was used as a performance metric.

Fig. 24: Comparison of volume overlap error for ten instances of the polyhedron pouring experiment. Time stepping with the standard contact model generated more error for larger time steps, while the error when using PEG was relatively imperceptible, set along the horizontal axis for all time steps tested.

Fig. 25: Numbers of contacts and SM-traps encountered at each time step for one polyhedron pouring simulation with time step of \( h = 0.01 \)s. Smaller time steps generated fewer SM-traps since the contact proximity parameter \( R \) was smaller.

as the time step increases. This is due to the fact that the time-stepping approach solves for impulses at the end of the time step based on normal directions determined at the beginning of the time step. As such, body rotations may generate unpredicted interpenetrations at the end of the time step, which are exacerbated given larger time steps.

In future work, we plan to explore efficiency and accuracy trade-offs associated with the design of heuristics used to define the set of active contacts, the size of time steps, and the number of primary potential contacts. When simulation speed is the main goal, it is desirable to minimize the size of the active set and the number of primary potential contacts. However, when high accuracy trumps speed, then multiple primary potential contacts per contact should be used and the active set would be larger. The pursuit of this work, will require a concomittant effort in the development of efficient, large-scale solvers of linear programs with complementarity constraints.

References


[D.E. Stewart and J. C. Trinkle, *An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction, Inter-


